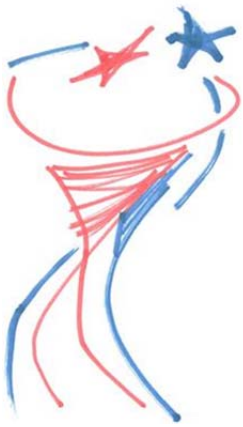
ICT - Information and Communication Technologies



PhoxTroT

Photonics for High-Performance, Low-Cost & Low-Energy Data Centers, High Performance Computing Systems: Terabit/s Optical Interconnect Technologies for On-Board, Board-to-Board, Rack-to-Rack Data Links

Collaborative Project
Grant Agreement Number: 318240

HPC and DataCentre Application Scenarios and Uses Deliverable 3.2

Deliverable number:	D3.2	**Work package number:**	3
Due date of deliverable:	30/09/2013 (M12)	**Actual submission date:**	11/11/2013 (M14)
Start date of the project:	01.10.2012	**Duration:**	48 months
Nature:	Report	**Dissemination level:**	CO

Lead beneficiary: Xyratex
Contact person: Richard Pitwon
Address: 1000-40 Langstone Technology Park
Phone: +44 (0)2392 49 6715
Email: Richard_pitwon@xyratex.com
Author(s): Richard Pitwon (Xyratex), Emmanouel Varvarigos (CTI), Sander Dorrestein (TE Connectivity), Kostas Christodoulopoulos (CTI), Apostolis Siokis (CTI)
Contributing beneficiaries: CTI, TE Connectivity

Abstract:
This document introduces application scenarios and uses for HPC and Datacentres in order to demonstrate how advanced high density optical routing architectures could be deployed into future Datacenter technology applications.
Keywords: Architectures, protocols, HPC, Datacentres

Project Information

PROJECT	
Project name:	Photonics for High-Performance, Low-Cost & Low-Energy Datacentres, High Performance Computing Systems: Terabit/s Optical Interconnect Technologies for On-Board, Board-to-Board, Rack-to-Rack data links
Project acronym:	PhoxTroT
Project start date:	01.10.2012
Project duration:	48 months
Contract number:	318240
Project coordinator:	Dr. Tolga Tekin - Fraunhofer
Instrument:	Large-scale integrating project - CP-IP
Activity:	ICT-8-3.5 - Core and disruptive photonic technologies

DOCUMENT	
Document title:	HPC and DataCentre application scenarios and uses
Document nature:	Report
Deliverable number:	D3.2
Due date of delivery:	30/09/2013 (M12)
Calendar date of delivery:	11/11/2013
Editor:	Tolga Tekin
Author(s):	Richard Pitwon (Xyratex), Emmanouel (Manos) Varvarigos (CTI), Sander Dorrestein (TE Connectivity), Kostas Christodoulopoulos (CTI), Apostolis Siokis (CTI
Lead beneficiary:	XYRATEX
Contributing beneficiaries:	CTI, TE Connectivity
Dissemination level:	CO
Work package number:	3
Work package title:	Architectures, Protocols & Design of HPC and DataCentre Interconnect Systems
Date created:	01/08/2013
Updated:	31/10/2013
Version:	3
Total number of pages:	38
Document status:	final

PU = Public ; PP = Restricted to other programme participants (including the Commission Services) ; RE = Restricted to a group specified by the consortium (including the Commission Services) ; CO = Confidential, only for members of the consortium (including the Commission Services)

Table of Contents

1 Executive Summary

This document introduces application scenarios and uses for HPC and Datacentres in order to demonstrate how advanced high density optical routing architectures could be deployed into future Datacentre technology applications.
A comprehensive analysis is provided of communication requirements and emergent paradigms for HPC and Datacentre applications. This is followed by a discussion on the challenges of incorporating optical interconnect into traditional data storage architectures, and how these challenges have been successfully addressed in recent internal demonstrations by Xyratex.

2 Introduction

In-system bandwidth densities driven by interconnect speeds and scalable I/O within data storage and server enclosures will continue to increase over the coming years thereby severely impacting cost and performance in future Datacentre systems. System embedded photonic interconnect technologies have been the subject of research and development for many years to provide a cost viable "eco-system" to mitigate this impending bottleneck.
This document introduces application scenarios and uses for HPC and Datacentres in order to demonstrate how advanced high density optical routing architectures could be deployed in future Datacentre applications.
We start by discussing HPC applications and their communication requirements. We initially report on related work on profiling and classifying HPC applications and then we present our experiments. We executed, monitored and analyzed a number of representative HPC applications, both at the server and the switches levels. We then turn our attention to Datacentres, where we report on related work on traffic analysis. A sub-section is dedicated to MapReduce traffic characteristics, a parallel processing paradigm that is increasingly being used for data-intensive applications in Cloud computing Datacentres.
The challenges of incorporating optical interconnect into traditional data storage architectures will also be considered with emphasis on the architectures defined by the Serial Attached SCSI protocol typically used in data storage subsystems and the recent successful demonstrations of optically enabled data storage platforms by Xyratex.
Finally, we also discuss how emerging commercial technologies, global disruptive research activities and international standards are influencing the commercial adoption of embedded optical interconnect into HPC and Datacentre platforms.

2.1 Document structure

The present deliverable is split into four major chapters:

- Introduction
- High Performance Computing
- Datacentres
- Optically enabled data storage platforms
- HPC and Datacentre systems evolution

2.2 Audience

This document is internal to PhoxTroT project consortium.

3 High Performance Computing (HPC) systems and applications

We first focus on High Performance Computing (HPC) systems. A key difference between HPC systems and Datacenters is that applications running on HPC systems typically get **dedicated access to the resources** (computation, network, storage), as opposed to the **sharing** and resource virtualization paradigm that is followed in Datacenters. So, in an HPC system one or several applications are running in a space shared manner, that is the users are granted a set of servers or racks or the whole system and each application is executed independent of the others or on the whole system and has dedicated access to the resources that it uses. Typically applications running on HPC systems (we will call these **HPC applications**) comprise tasks that run on processors in a distributed/parallel manner and communicate through messages. MPI (Message Passing Interface Standard) has become the "industry standard" for writing message passing programs on HPC platforms. So for HPC systems we will focus on applications that follow the MPI programming paradigm.

A classification of HPC applications was performed in [1], where 13 "dwarfs" or "kernels" were defined. Kernels are algorithmic methods that represent a wide spectrum of computations for various application domains. In particular the kernels defined in [1] were as follows: Dense Linear Algebra, Sparse Linear Algebra, Spectral Methods, N-Body Methods, Structured grids, Unstructured grids, Monte Carlo, Combinational Logic, Graph traversal, Dynamic Programming, Backtrack and Branch&Bound, Construct Graphical Models, Finite State Machine. Distributed algorithms of a particular kernel class can be implemented differently, but will exhibit similar communication patterns. Real applications are typically composed of a single or a number of kernels, that can be executed in a time-shared manner (on a common set of processors) or space-shared manner (each kernel uniquely occupying a subset of available processors). Thus, studying these kernels and examining their communication patterns so as to optimize them and serve the created traffic in an efficient can boost the performance of the system.

In the same direction, [2] profiled 8 HPC applications using a MPI profiling tool called IPM (Integrated Performance Monitoring) [3]. IPM profiles performance aspects and resource utilization of a parallel program, maintaining a low-overhead. Using this tool the authors of [2] provided detailed information on the communication requirements and communication patterns exhibited by the examined applications. In particular, the applications that were examined are listed in Table 1. They range from simple kernels to more complex applications.

Name	Discipline	Problem and Method	Structure
BBeam3D	High Energy Physics	Vlasov-Poisson via Particle in Cell and FFT	Particle/Grid
Cactus	Astrophysics	Einstein's Theory of GR via Finite Differencing	Grid
GTC	Magnetic Fusion	Vlasov-Poisson via Particle in Cell	Particle/Grid

LBCFD	Fluid Dynamics	Navier-Stokes via Lattice Boltzmann Method	Grid/Lattice
MADbench	Cosmology	CMB Analysis via Newton-Raphson	Dense Matrix
PARATEC	Material Science	Density Functional Theory via FFT	Fourier/Grid
PMEMD	Life Sciences	Molecular Dynamics via Particle Mesh Ewald	Particle
SuperLU	Linear Algebra	Sparse Solve via LU Decomposition	Sparse Matrix

Table 1: Short description of applications examined in [2].

Table 2 shows the *MPI communication call types and call counts* for the applications examined in [2]. To read this table, one has to be accustomed to how MPI message passing works. We assume an application that comprises tasks that run on several processors in a parallel manner. The messages exchanged between the processors-tasks following the MPI message passing standard are categorized into 2 classes: (a) **point-to-point** and (b) **collectives**:

- *Point-to-point (PTP) communication routines (involve message passing between two, different MPI tasks)*. *Send, Recv*: Basic blocking send and receive operation. Routine returns only after the application buffer in the sending task is free for reuse. *Isend*, *Irecv*: non-blocking send, recv respectively. Processing continues immediately without waiting for the message to be copied out from the application buffer. *Wait, Waitall, Waitany*: Wait blocks until a specified non-blocking send or receive operation has completed. For multiple non-blocking operations, the programmer can specify all (Waitall) or any (Waitany) completions. *Sendrecv*: Will block until the sending application buffer is free for reuse and until the receiving application buffer contains the received message. *Test*: MPI Test checks the status of a specified non-blocking send or receive operation.
- *Collective communication routines (involve all processes).* *Gather*: Gather distinct messages from each task in the group to a single destination task (AllGather: to all tasks in a group). *Reduce*: Applies a reduction operation on all tasks in the group and places the result in one task (AllReduce: in all tasks). *Bcast*: Broadcasts (sends) a message to all other processes in the group. *Barrier*: Creates a barrier synchronization in a group. Each task, when reaching the Barrier call blocks until all tasks in the group reach the same Barrier call. *AllGatherv*: Same as Allgather but allows for messages to be of different sizes and displacements. *Alltoall*: sends a distinct message from each process to every other process (*Alltoallv*: Same as Alltoall but allows for messages to be of different sizes and displacements).

According to the findings of [2], presented in Table 2, most applications utilized mostly point-to-point communication routines (over 90% of all MPI calls), except for GTC (which used mainly Gather – which is a collective operation). Non-blocking communication was the predominant point-to-point communication mode.

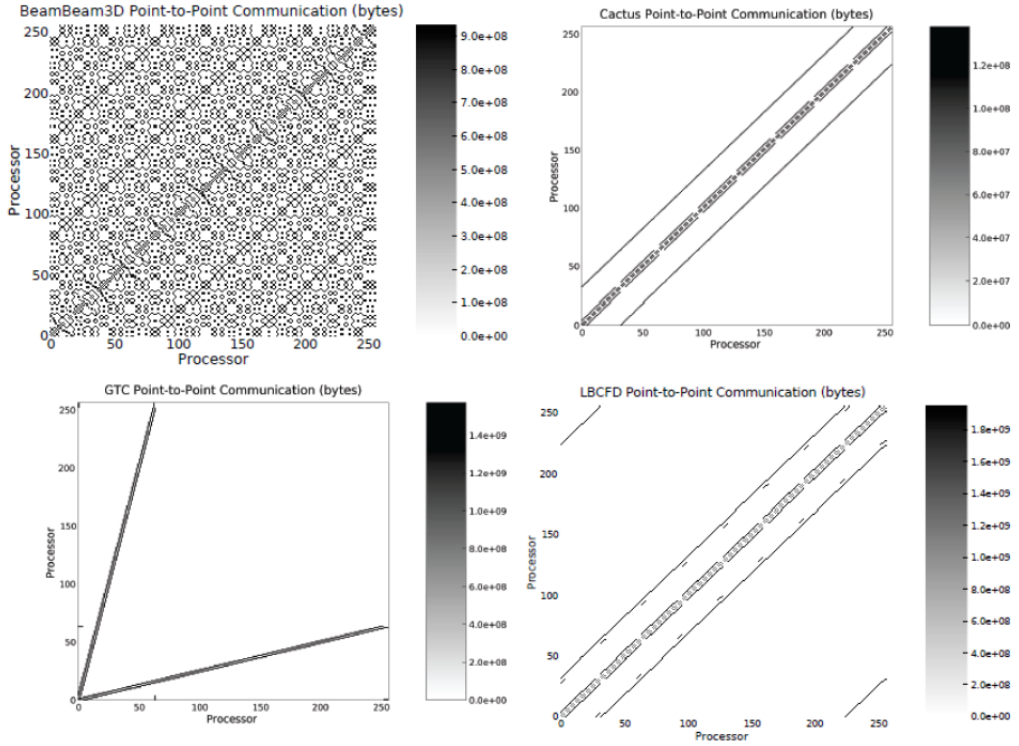
Function	BB3D	Cactus	GTC	LBCFD	MADbench	PARATEC	PMEMD	SuperLU
Isend	0%	26.8%	0%	40.0%	5.3%	25.1%	32.7%	16.4%
Irecv	33.1%	26.8%	0%	40.0%	0%	24.8%	29.3%	15.7%

Wait	33.1 %	39.3 %	0%	0%	0%	49.6%	0%	30.6%
Waitall	0%	6.5%	0%	20. 0%	0%	0.1%	0.6%	0%
Waitany	0%	0%	0%	0%	0%	0%	36.6%	0%
Sendrecv	0%	0%	40.8 %	0%	30.1%	0%	0%	0%
Send	33.1 %	0%	0%	0%	32.2%	0%	0%	14.7%
Gather	0%	0%	47.4 %	0%	0%	0.02%	0%	0%
(All)Reduce	0.5%	0.5%	11.7 %	0.02%	13.6%	0%	0.7%	1.9%
Bcast	0.02 %	0%	0.04 %	0.08%	6.8%	0.03%	0%	5.3%
% PTP Calls	**99.2 %**	**98.0 %**	**40.8 %**	**99.8%**	**66.5%**	**99.8%**	**97.7%**	**81.0%**
TDC (max, avg)	**66, 66**	**6, 5**	**10, 4**	**6, 6**	**44, 39**	**255, 255**	**255, 55**	**30, 30**
FCN Utilization	**25.8 %**	**2.0%**	**1.6%**	**2.3%**	**15.3%**	**99.6%**	**21.4%**	**11.7%**

Table 2: %MPI communication calls. % of point-to-point (PTP) messaging, maximum and average TDC (Topological Degree of Communication) thresholded by 2 KB, FCN (Fully Connected Network) utilization (thresholded by 2 KB) for 256 processors. Results are taken from [2].

The authors in [2] then turn their attention to the *message sizes* used by applications. The Bandwidth*Delay (latency) product describes how many bytes must be "in-flight" to fully utilize the available link bandwidth. If message size is greater than bandwidth*delay product then it is a **bandwidth bound** message (channel saturation), else **latency bound** (these messages cannot be speeded up by increasing the available bandwidth). In [2], 2 KB was chosen as a threshold (after examining the bandwidth*delay of several HPC interconnects), meaning that messages larger than 2 KB are considered bandwidth bound. According to the findings, collective communication requirements differ from point-to-point: 90% of the collective messages were 2 KB or less (latency bound messages). Note that IBM's BlueGene SuperComputer has a separate low-bandwidth network for collective operations. On the other hand, point-to-point messages were observed to be mainly bandwidth bound. For all but two of the applications examined, message sizes larger than 2 KB accounted for more than 75% of the overall point-to-point message sizes.
Finally, the authors in [2] also presented the **logical communication graphs** of the examined applications. A logical communication graph expresses the amount of data that is exchanged between processors throughout the application execution. The term *logical* implies that these graphs are related to the program structure, that is, they do not depend on the system that are executed and on the underlying interconnect. The logical communication graphs of the examined applications for P=256 processors are depicted in Figure 1. The x- and y-axis in these graphs correspond to the processes, and there is a dot in the graph if the x-axis process communicated with the related y-axis process. Cactus, GTC, LBCFD applications was observed to exhibit low Topological Degree of Communication (TDC). The TDC shows the number of peers, the number of processes that

each process communicates. Note that this reflects the point-to-point messages, since collective operations involve all the processes. As mentioned above, Cactus, GTC, LBCFD applications have a low TDC, which is much lower than what a Full Bisection Bandwidth (FBB) network would provide. BB3D, PMEMD, SuperLU, MADbench exhibited higher TDC than Cactus, GTC and LBCFD, but still significantly less than the number of links a FBB network would provide. PARATEC (that is based on a Fast Fourier Transformation –FFT-kernel) was the only application examined that would utilize the efficiently the bisection bandwidth provided by a FBB. All other applications would underutilize a FBB interconnect. Building network to provide Full Bisection Bandwidth (FBB) is not scalable for large numbers of processors, since the cost of a fat-tree FBB network increases super-linearly to the number of processors. Moreover, the bisection bandwidth offered in a FBB is mostly underutilized for a broad range of applications, as observed above, FBB networks are not a preferred solution for current and future HPC environments [2], [4].



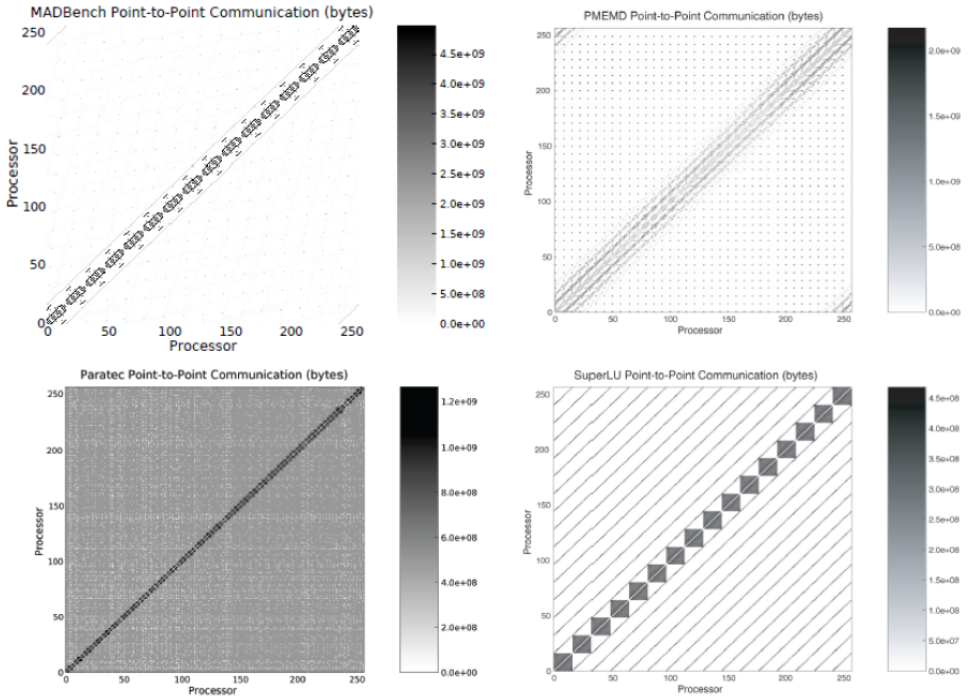


Figure 1, courtesy of [2]**: Topological Connectivity of the applications for P = 256 processors (ignoring latency-bound messages).**

Finally, regarding HPC traffic, textbook [4] has a collection of synthetic traffic patterns in the form of spatial distribution of messages. These distribution functions (see Table 3) give the destination address *d* (or bits of the destination address) as a function of the source address *s* (or bits of it). Several of these patterns are based on communication patterns that arise in particular kernels or applications. For example, shuffle permutation arises in Fast Fourier Transformation-FFT- or sorting. Fluid dynamics often exhibit neighbor patterns. Random traffic is described by a traffic matrix Λ, with all entries $\lambda_{sd} = 1/N$, where N is the number of processors. In permutation traffic all traffic from each source s is directed to one destination, represented by a permutation function $\pi(s)$ that maps source s to a particular destination d. Bit permutations are those in which each bit d_i of the b-bit destination address is a function of one bit of the source address s_j where j is a function of i. Digit permutations (for torus & butterflies) are similar to bit permutations but for addresses expressed as digits.

Name	Pattern
Random	$\lambda_{sd} = 1/N$
Permutation	$d = \pi(S)$
Bit permutation	$d_i = s_{f(i)} \oplus g(i)$
Bit complement	$d_i = \neg s_i$
Bit reverse	$d_i = s_{b-i-1}$
Bit rotation	$d_i = s_{i+1} mod\ b$
Shuffle	$d_i = s_{i-1} mod\ b$
Transpose	$d_i = s_{i+b/2} mod\ b$
Digit permutations	$d_x = f(s_{g(x)})$
Tornado	$d_x = s_x + (\lceil k/2 \rceil - 1) mod\ k$

Neighbor	$d_x = s_x + 1 \; mod \; k$

Table 2: A collection of synthetic traffic patterns for HPC kernels and applications, according to textbook [4].

Since the above corresponds to point-to-point messages, there are also textbook models for collective communication:

- Single Node and Multinode Broadcast: In Single Mode Broadcast communication task, the same packet is sent from a processor to every other processor. In Multinode Broadcast a Single Node Broadcast is performed simultaneously from all nodes.
- Single Node and Multinode Accumulation: In Single Mode Accumulation communication task, a packet is sent from all nodes to a given node. The Multinode Accumulation involves a separate single node accumulation at each node.
- Single Node Scatter, Single Node Gather and Total Exchange: In Single Node Scatter communication task, involves sending a separate packet from a single node to every other node. A dual task called Single Node Gather, involves collecting a packet at a given node from every other node. In Total Exchange communication task, every node sends a (different) packet to every other node in the network.

The aforementioned communication tasks form a hierarchy in terms of difficulty (Figure 2), in the sense that an algorithm solving one problem in the hierarchy can also solve the next problem in the hierarchy in no additional time.

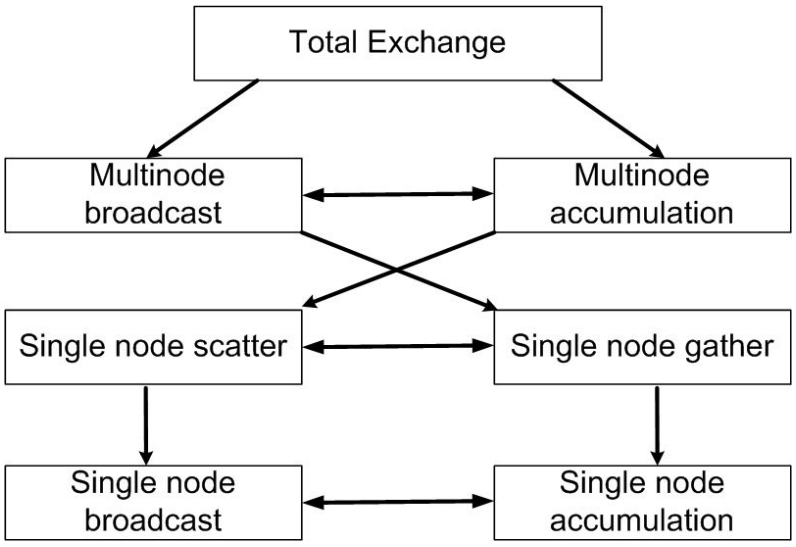


Figure 2. Hierarchy of basic collective communication in interconnection networks. A directed arc from problem A to problem B indicates that an algorthm that solves A can also solve B, and that the optimal time for solving A is not more than the optimal time for solving B. A horizontal bidirectional arc indicates a duality relation.

3.1 Process-level HPC applications profiling

We executed a number of HPC applications on HellasGrid cluster HG-06-EKT and profiled their execution. The HPC applications were run for 480 MPI-ranks (a rank was pinned to a processor), but were also tested for fewer ranks. We profiled three representative HPC applications. In particular, we installed POP [5], SuperLU [6], FFTW [7] applications and profiled using IPM [3]. These applications exhibit different communication behaviours. POP is an application with high traffic locality. In SuperLU global communication takes place, but it is data intensive mainly locally. FFTW is an application that exhibits global communication (all ranks communicate point-to-point with all other ranks) that is globally traffic intensive.

3.1.1 POP application

The Parallel Ocean Program (POP) [5] was specifically developed to take advantage of high performance computer architectures and optimized particularly for toroidal topology interconnects. POP is a three-dimensional ocean circulation model designed primarily for studying the ocean climate system. The model solves the two or three-dimensional primitive equations for fluid motions under hydrostatic and Boussinesq approximations.
POP is an application with a relatively high local communication. Typically, a process (rank) is assigned an area of the fluid and exchanges messages with its spatially adjacent processes. So in a two dimensional experiment a process exchanges messages mainly with four (north-south-east-west) processes, while in three dimensional experiments the number of peers is 6.

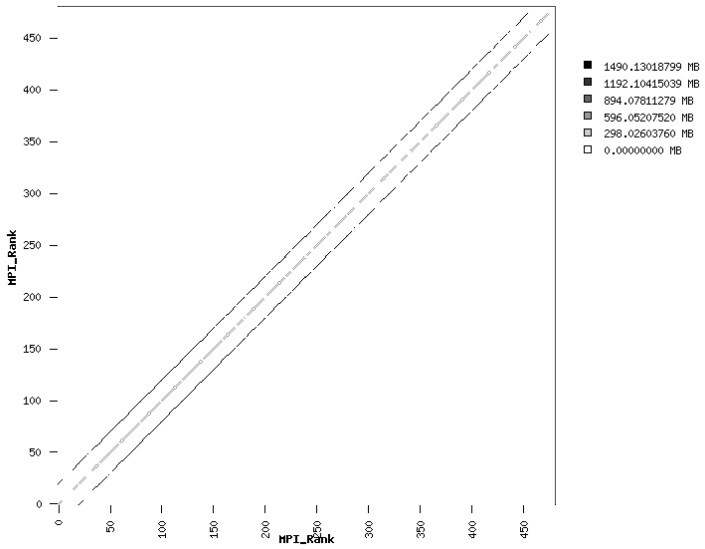


Figure 3. Logical communication graph captured by running POP application for 480 mpi-ranks.

Figure 3 shows the logical communication graph of a two dimensional POP experiment using 480 MPI-ranks (processes). This graph depicts the point-to-point communication aggregated over the execution of the whole application, that is the total size of messages exchanged throughout the application execution (as done in Figure 1). For 480 MPI-ranks we observed the average Topological Degree of Communication (TDC) to be 5.72. Process 0 communicated point-to-point with all other process, so it has communication degree equal to 479, but the related volume was very low (process 0 distributes some input

parameters once at the beginning of the application). All the rest ranks communicate with 2, 3, 4 or 5 ranks, with 4 (north-south-east-west) being the most dominant peers in this two dimensional POP experiment.
For the POP experiment with 480 ranks, 31.72% of total MPI time is collective communication, and 66.63% point-to-point (table 3). The remaining percentage corresponds to MPI environment management routines. Figure 4 shows the CDFs of the sizes of the point-to-point and the collective messages. For collective communication: 99.99% of the message sizes are less than or equal to 320 Bytes, which is very low. For point-to-point communication 65% of the message sizes are greater than 2KB and 99.99 < 9 KB. So there are some relatively large point-to-point messages.

Function	Percentage of MPI time
Isend	0.09%
Waitall	66.54%
(All)Reduce	27.89%
Bcast	3.12%
Barrier	0.71%

Table 3: Percentage of time spent on specific MPI communication calls for POP application.

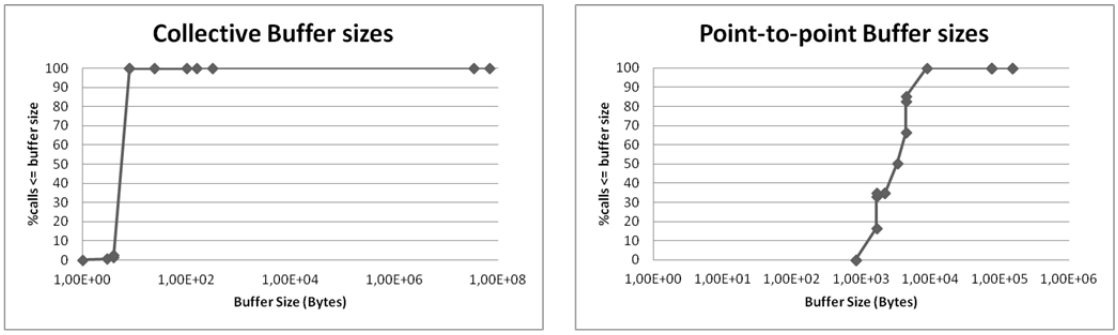


Figure 4. Buffer sizes distribution for collective and point-to-point communication for POP application executed on 480 MPI-ranks

We observed that 95.1% of total number of messages correspond to point-to-point communication, while 4.9% to collective. In terms of volume, 97.58% volume of traffic is produced by point-to-point messages and 2.42% by collectives.
So, from our analyses we observed that in POP application the majority of traffic is point-to-point. The majority of point-to-point messages and all collective messages are small, but there are few point-to-point messages that are relatively large. However, the point-to-point messages are exchanged between "adjacent" nodes. Thus, POP application has high traffic locality.

3.1.2 SuperLU application

SuperLU [6] is a general purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines. The library routines will perform an LU decomposition (or factorization) with partial pivoting and triangular system solves through forward and back substitution. LU decomposition factors a matrix as the product of a lower triangular matrix (L) and an upper triangular matrix (U) which can be viewed as the matrix form of Gaussian elimination. A wide number of scientific applications perform LU decompositions to solve square systems of linear equations, to invert a matrix, or to compute its determinant.

SuperLU is an application where global communication takes place. Figure 5 presents the logical communication graph of an execution of SuperLU with 480 MPI-ranks to decompose the "webbase" sparse matrix as found in the University of Florida sparse matrix collection [8], which corresponds to an instance of the connections of the World Wide Web (WWW). This particular matrix is square with size 124,651 x 124,651, is integer, nonsymetric, and has 207,214 non-zero elements. In this experiment, the average and maximum topological degree of communication (TDC) is 479, which means that every process (MPI-rank) talks point-to-point with every other process. However, every process exchanges a large portion of data only with a subset of the MPI-ranks in the network, and in particular with 50 processes in this particular experiment. Take Figure 5 and e.g. MPI-rank *i=10* as the source. The destination-ranks that rank *i=10* sends messages is found by taking a vertical cut at x-axis=10. Although according to the log files the 10th rank sends messages to all other ranks, apart from 50 most dominant ranks for the rest the traffic is so low that is not shown (it is white) in this figure. So in the figure we can only see the 50 ranks that are the dominant ones. We see that rank *i=10* talks mainly with its directly adjacent ranks 1 to 12 apart from rank-10 (it doesn't talk to itself), this corresponds to the square structures in the logical communication graph of Figure 5. Also it talks to some MPI-rank that are more distant, and in particular to ranks $i + n*13$, n=1,2,….39. As a general comment, we see that the communication graph is well structured, and this comes from the particular implementation of SuperLU and not on the traffic matrix that we factorized.

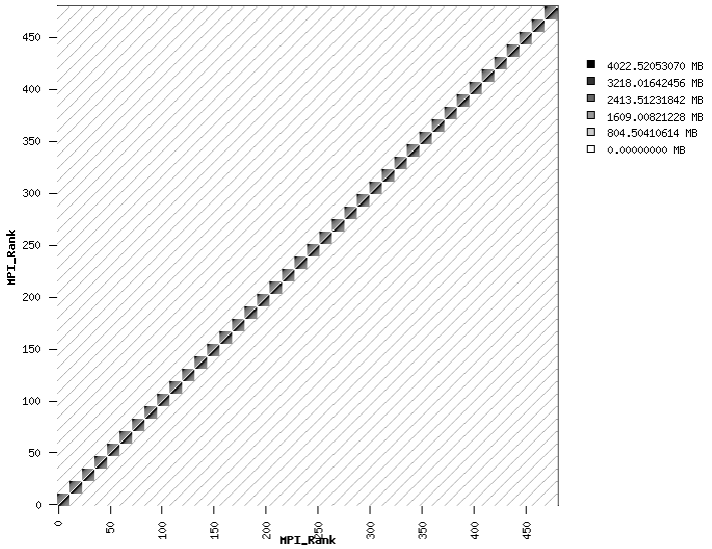


Figure 5. Logical communication graph captured by running SuperLU application for 480 mpi-ranks

For this experiment of SuperLU, 25.83% of total MPI time is collective communication, and 70.66% point-to-point (table 4). The remaining communication percentage corresponds to MPI environment management routines.

Function	Percentage of MPI time
Isend	0.39%
Irecv	0.05%
Recv	11.58%
Wait	41.07%
(All)Gather(v)	3%

(All)Reduce	3.15%
Bcast	6.2%
Barrier	0.05%
Alltoall(v)	13.43%
Test	17.57%

Table 4: MPI communication calls for SuperLU application.

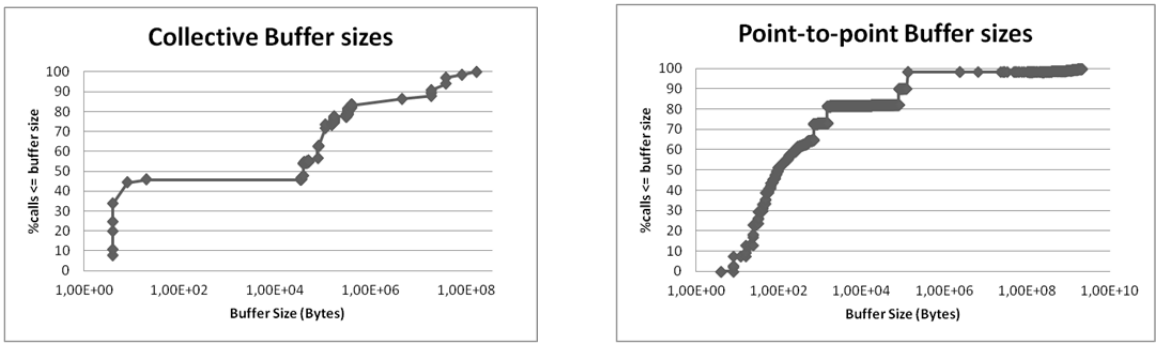


Figure 6. Buffer sizes distribution for collective and point-to-point communication for SuperLU application

Figure 6 shows the CDFs of the sizes of the point-to-point and the collective messages. For collective communication: 45.8% of the message sizes are less than or equal to 20 Bytes. For point-to-point communication 81.67% of the message sizes are less than 2KB. Moreover, 99.99% of total number of messages corresponds to point-to-point communication. In terms of volume, 99.99% of total traffic volume is generated by point-to-point messages.

As a conclusion we see that the traffic of SuperLU is mainly point-to-point, since the number of collective calls is very low and their volume is negligible. Point-to-point traffic is mainly directed to adjacent nodes and some distant nodes, having a well-structured communication pattern.

3.1.3 FFTW application

FFTW [7] is an implementation of the discrete Fourier transform (DFT) that adapts to the hardware in order to maximize performance. It represents the class of codes where global and data intensive communication takes place.

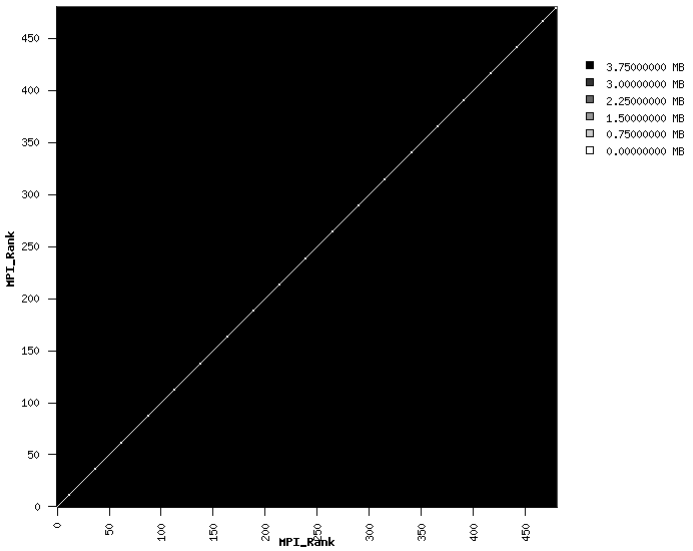


Figure 7. Logical communication graph captured by running FFTW application for 480 mpi-ranks.

For 480 MPI-ranks that we executed FFTW, we observed that the average and maximum TDC of point-to-point communication is 479 (figure 7). This means that every rank talks point-to-point with every other rank. Applications with such communication behaviour require high bisection bandwidth, which is one of the drawbacks of low-degree torus networks. Note that FFT is very widely used, and although its communication graph changes depending on the algorithm implementation it is considered in general as one of the applications that requires high bisection bandwidth and thus a full bisection bandwidth (FBB) network is appropriate. This fact lead Cray to replace the, not suitable for global communication patterns, 3D Torus architecture of XT and XE series with a more efficient architecture (Dragonfly) for their next generation HPC systems (XC series), to be able to execute a wider range of applications [9].

Function	Percentage of MPI time
Sendrecv	97.39%
Allreduce	2.60%

Table 5: MPI communication calls for FFTW application

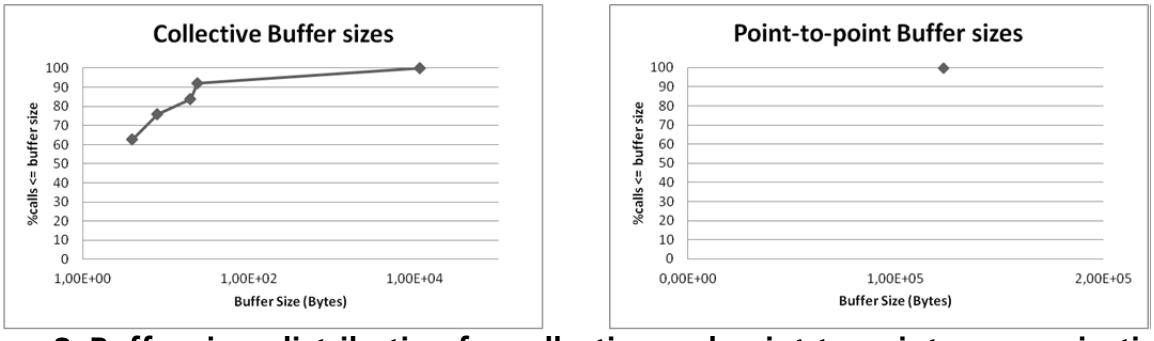


Figure 8. Buffer sizes distribution for collective and point-to-point communication for SuperLU application

For FFTW, all MPI messages are either Sendrecv (point-to-point call) or Allreduce (collective call). We measured 97.39% of total MPI time to be point-to-point (table 5). Figure 8 shows

the CDFs of the sizes of the point-to-point and the collective messages. All sendrecv messages (the only point-to-point messages in this experiment) are 120 Kbytes. Allreduce messages (the only collective messages in this experiment) vary but all of them are less than 1KByte. We observed that 98.73% of the total number of messages to correspond to point-to-point communication. Finally, in terms of volume 99.99% of total traffic volume is generated by point-to-point messages.
As a conclusion we see that the traffic of FFTW is mainly point-to-point and in particular sendrecv messages. However, these point-to-point messages are exchanged between all pairs of ranks/processors. Thus communication is global and a network that provides high bisection bandwidth is needed to support this application.

3.2 Server-level HPC applications profiling

We executed the same applications (POP, SuperLU, FFTW) on 16 servers - 64 ranks (fewer than in the previous experiments – due to issues with accessing the network switches) and we monitored the traffic that crossed the switch that is connected to these servers. The switch was an Ethernet 1Gbps switch. We used sFlow [10] to monitor the traffic that crosses through the switch and created traffic matrices that represent the traffic that was transferred among the 16 servers. Monitoring information at the switch was logged every 1 sec, but in the traffic matrices that we present in the following we accumulated the traffic over 5 sec.
The main difference between the **traffic matrices** that we present in this section and the logical communication graphs that were presented in the previous section is that in the logical communication graphs the x- and y-axis of the communication graphs correspond to mpi-ranks, while in the traffic matrices the x- and y-axis correspond to servers. Typically, an mpi-rank is pinned to a processor and more and more and more processors are integrated into the servers (4 in the system that we used for our experiments). Moreover, there can be many ways to allocate ranks to processors. The logical communication graph is independent of this allocation (and this is why the term "logical" is used), while the traffic matrix is created for a particular allocation and thus reflects this choice. In particular, for the traffic matrices that we present in this section we used mpich2's default allocation of ranks to processors, which allocates sequentially, in increasing order, the ranks to the available processors. Another difference has to do with the type of information that the graphs represent. The logical communication graphs presented in the previous section show the total volume of point-to-point traffic that was carried between ranks/processes throughout the execution of the whole application, that is they do not convey temporal information (to be more specific they have reference the running time of the whole application, which is different for each execution). On the other hand, the traffic matrices that we present in this section correspond to traffic for a specific duration of time, so they convey temporal information. In particularly we monitored the traffic that crossed the switch, which the servers executing the application were connected to, every 1 sec, but in the traffic matrices that we present here we accumulated 5 consecutive ones, so the traffic matrices presented correspond to 5 sec. We choose the time period of 5 sec because some traffic patterns appeared more vividly at this scale, while they were not very visible in traffic matrices of 1 sec. Finally, note that the logical communication graphs capture only the point-to-point MPI calls, while in the traffic matrices we show here both point-to-point and collective calls contribute.

3.2.1 POP application

Figure 9 shows the traffic matrices for POP application (on 16 servers - 64 ranks). Similar to the communication graph of the same application, we see that the application has strong locality. A server (as opposed to a rank/process in the logical communication graph case) communicates manly with its adjacent servers, and global traffic is very limited (almost none).

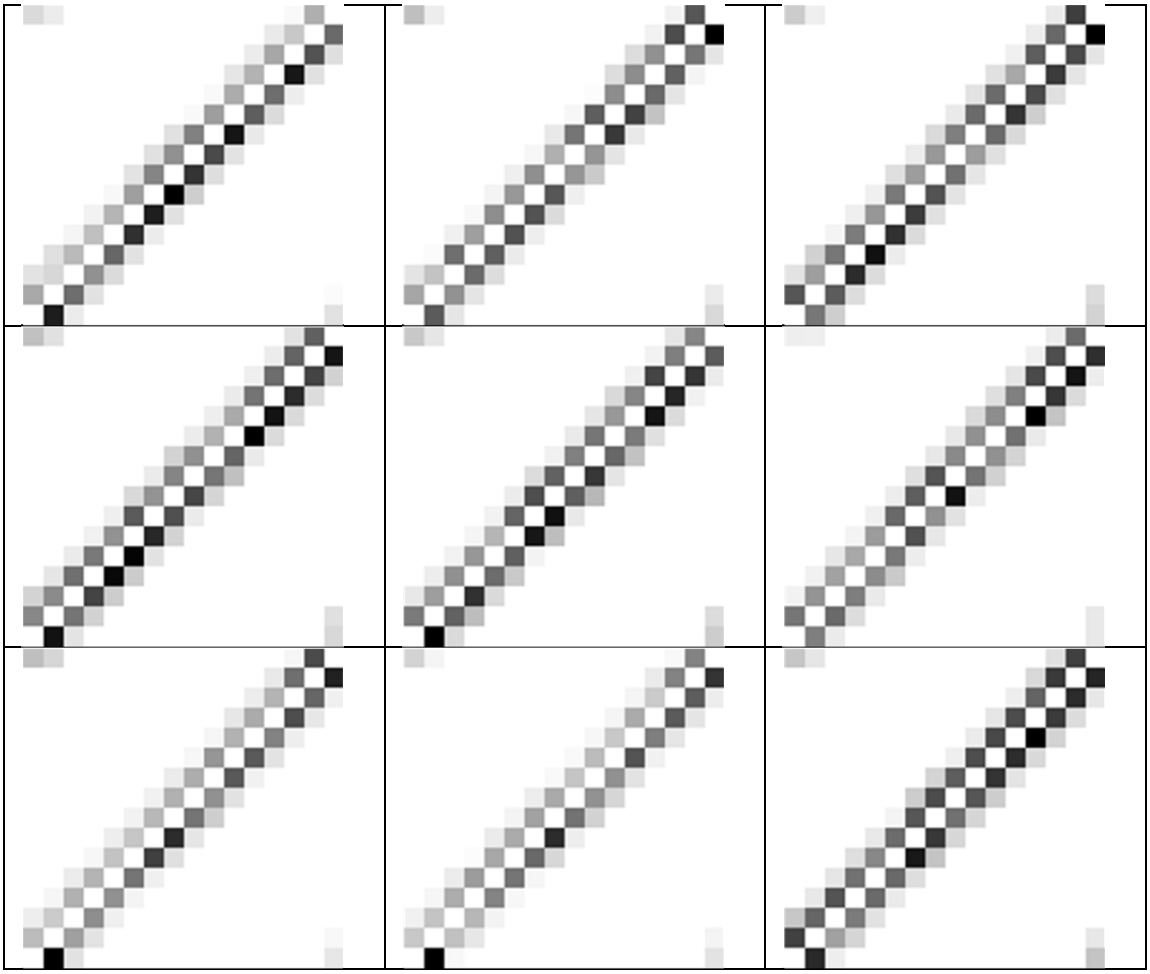


Figure 9: Traffic matrices obtained by running POP application on 16 servers and monitoring the traffic that crosses the switch. Matrices correspond to traffic accumulated over 5 sec.

3.2.2 SuperLU application

Figure 10 shows the traffic matrices for SuperLU application. Similar to the logical communication graph of the same application, we see that there is global traffic, but the traffic is much heavier between adjacent servers.

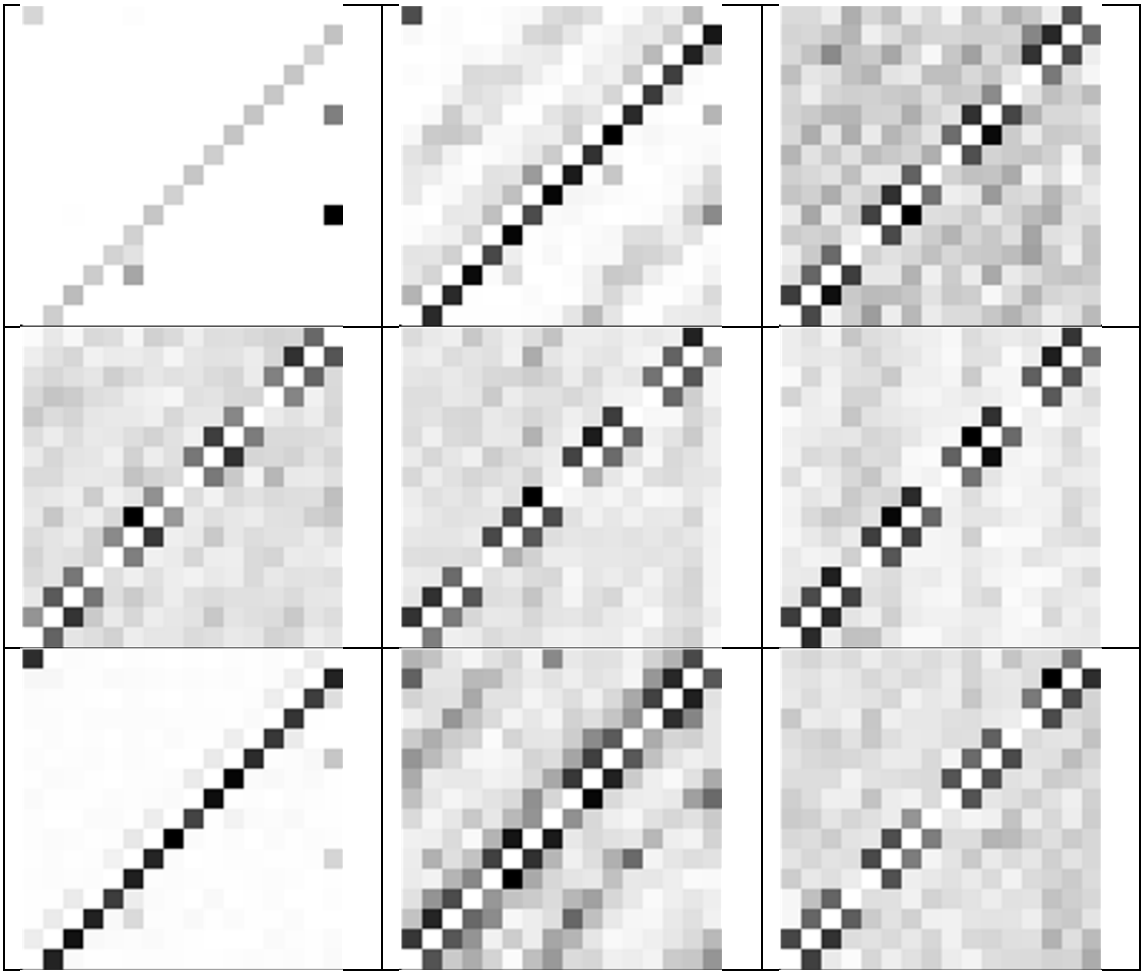
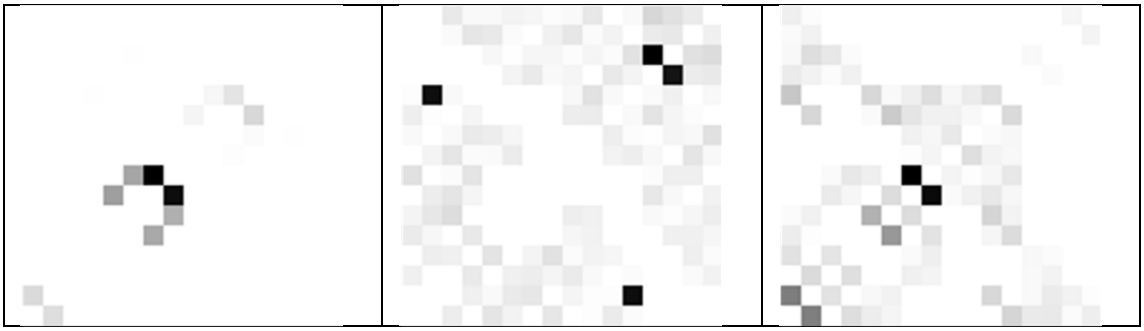


Figure 10. Traffic matrices obtained by running SuperLU application on 16 servers and monitoring the traffic that crosses the switch. Matrices correspond to traffic accumulated over 5 sec.

3.2.3 FFTW application

Figure 11 shows the traffic matrices for FFTW application. Similar to the communication graph of the same application, we see that there is global communication and almost no locality. There are some hot-spots, that most times are different at different time instances, but communication occurs between all servers.



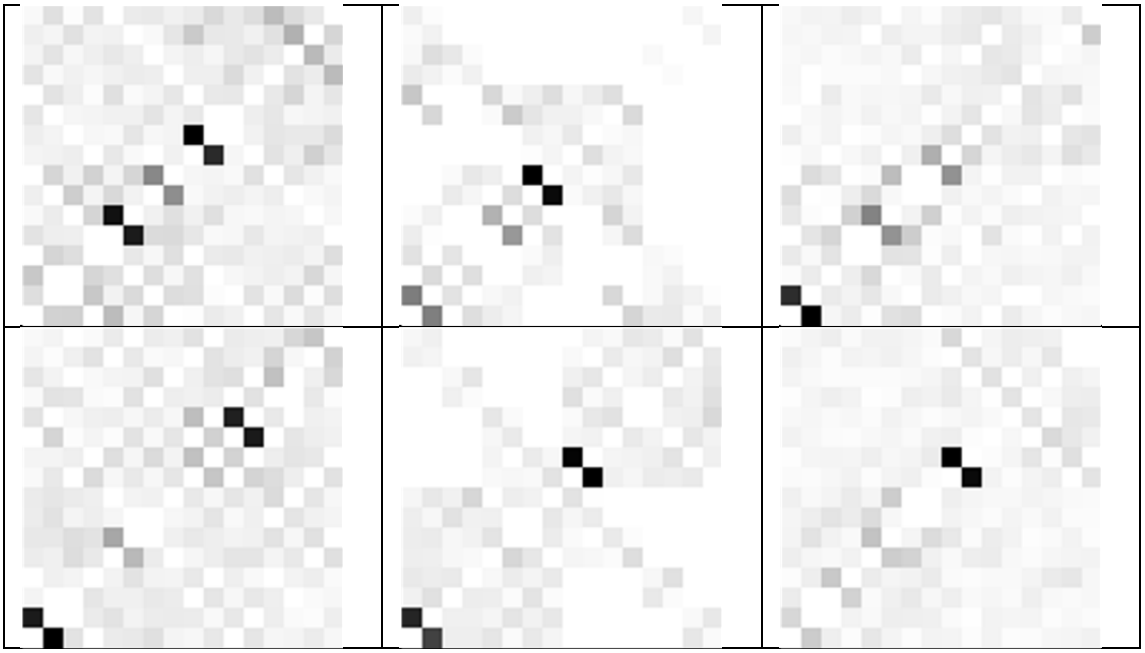


Figure 11: Traffic matrices obtained by running FFTW application and monitoring the traffic that crosses the switch. Matrices correspond to traffic accumulated over 5 sec.

4 Datacentres systems and applications

Datacenters provide computation and storage equipment to meet the data processing and storage requirements of an organization or for various users when we consider Cloud Datacenter. So the requirements can vary strongly from organisation to organization and from user to user, as opposed to HPC systems that are mainly used to run HPC and scientific applications. In Cloud Datacenters in many cases the resources are virtualized and shared as an infrastructure, platform or software as a service manner, so in this case we are talking about a multi-tenant environment where all resources are shared in a dynamic manner.

Most of the current Datacentres are based on commodity switches to build the interconnection network. The network is usually a canonical fat-tree 2-Tier or 3-Tier architecture, most times built with Ethernet switches. Typically incoming and outgoing traffic (communication to the outside world – internet) enters/exits at/from the root of this tree network. Full bisection bandwidth (FBB) fat-trees or, most commonly, oversubscribed fat-trees to reduce the cost are typically used to build the Datacenters.

Figure 12 presents a 3-Tier fat-tree topology. The trees are created as follows. The servers are mounted into racks and are connected through a Top-of-the-Rack Switch (ToR – edge or 1st tier) using 1 or 10 Gbps links. These ToR switches are further inter-connected through Aggregate Switches that comprise the 2nd tier, using 10 Gbps or 40 Gbps links. In 3-Tier topologies one more level is added in which the Aggregate Switches are interconnected using the Core Switches either at 10 Gbps or 40 Gbps links to form the fat-tree. Employing low rate redundant switches at the Aggregation and Core tiers, so that e.g. each ToR switch is connected to more than one Aggregation Switch and in turn each Aggregation Switch is connected to more than one Core Switch and multi-path routing as proposed in [11], can be used to increase the bandwidth of the fat-tree and provide up to FBB at the servers.

The Datacentres can be categorized in three classes: university campus, private enterprise and Cloud-computing Datacentres. In [13] SNMP data from 19 corporate and enterprise Datacentres and packet traces from one of the examined Datacentres were collected and analyzed. In [12] 10 Datacentres were profiled (3 university campus, 2 private enterprise and 5 commercial Cloud). In [14], the authors collected and analyzed 2 months data from a Cloud Datacentre that runs MapReduce jobs. The authors deduced that there are some common traffic characteristics in the different Datacentres (e.g. average packet size) while other characteristics (e.g. applications) vary significantly between the Datacentre categories. In what follows, we present an overview of the findings that were presented in the above referenced papers, since obtaining traces from operational Datacenters is quite difficult due to NDA agreements between Datacenter operators and users/organizations that use them.

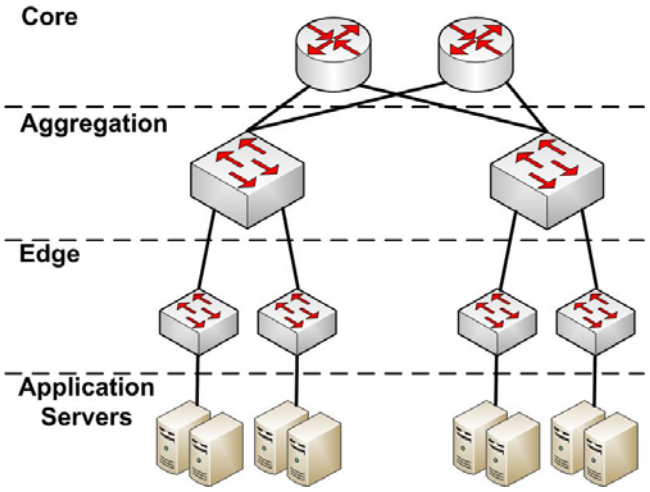


Figure 12: Typical interconnection network architecture of a Datacentre

4.1.1 Applications

The applications running on Datacentres vary among the different Datacentre categories but also among Datacentres of the same category. According to [12] campus Datacentres were observed to utilize the network mostly for distributed file systems traffic (e.g. AFS, NCP), but there were others where the majority of traffic was HTTP, HTTPS mixed with other applications such as file sharing (SMB). In private Datacentres a mix of HTTP, HTTPS, LDAP and custom applications was observed. Finally, some Cloud Datacentres were utilized for running MapReduce jobs, while others for hosting a variety of applications, ranging from messaging and Webmail to Web portals.

4.1.2 Flow duration, size, inter-arrival rates

A traffic flow is defined as an active connection (usually TCP) between 2 or more servers. It was observed in [12] that most traffic flows were small – less than 10 KB for all Datacentres and a significant fraction of these lasted under a few hundreds of milliseconds. The inter-arrival times of the flows were observed to be less than 10μs for 2 – 13% of the flows. In general, campus Datacentres were observed to have less churn than private and Cloud Datacentres [12], [13], [14] (for the private Datacentre examined in [13] 80% of the flows had inter-arrival times less than 1ms, while for the three campus Datacentres 80% of the flows had inter-arrival times between 4ms and 40ms). Finally, although the distribution

of the number of active flows was observed to vary across the Datacentres, the number of active flows at a switch at any given interval was less than 10000.

4.1.3 Packet sizes and inter-arrival rates

It was observed in [13] that the packet sizes exhibited a bimodal pattern, with most packet sizes clustering around 200 and 1400 bytes. The small packets were application keep-alive packets and TCP acknowledgments. The 1400 packets were parts of large files fragmented to the maximum packet size of the Ethernet networks. Packet arrivals were observed to exhibit an ON/OFF pattern. ON/OFF period and inter-arrival rates follow heavy-tailed distributions (Lognormal, Weilbul). Specifically, for private Datacentres ON periods and inter-arrival packet rates were modeled to follow the Lognormal distribution, while for campus Datacentres they were modeled to follow the Weilbul distribution.

4.1.4 Traffic flow locality

Traffic flow locality describes if the traffic generated by the servers in a rack is directed to the same rack (intra-rack traffic) or if it is directed to other racks (inter-rack traffic). Figure 13 reproduces the findings of [12]. It was observed that in campus and private Datacentres the traffic flow ratio for intra-rack traffic ranged from 10% to 40%. On the other hand, in Cloud Datacentres 80% of the traffic was observed to be intra-rack. This is probably due to better placement of the dependent services – servers that exchange high traffic between each other are located into the same rack by the network operators.

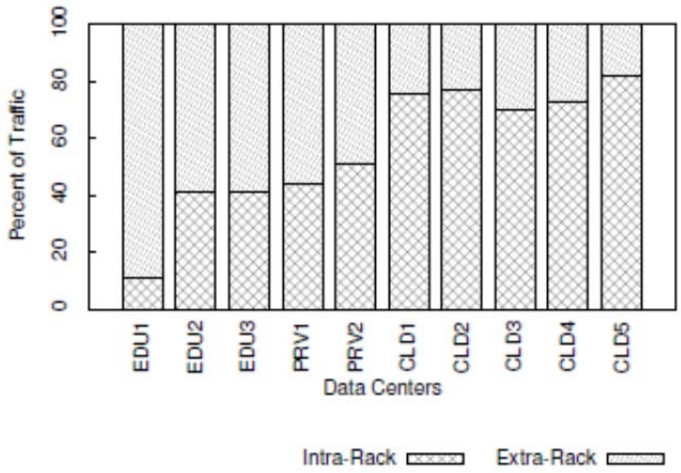


Figure 13, courtesy of [12]**: Intra-rack and inter-rack ratios for 10 Datacentres.**

4.1.5 Link utilization

In all kinds of Datacentres the link utilization inside the rack and at the Aggregate level was observed in [12] to be quite moderate, while the utilization on the Core level was high. It was also observed that only a small fraction of the existing bisection capacity (the aggregate capacity of the top-tier links - core) was utilized within a given time interval in all the examined Datacentres, but when load increased hot-spots appeared. These hot spots appeared at the Core level, and congestion, in most of the Datacentres, was occasional. In some of the Cloud Datacentres, a significant fraction of core links was observed to congest (create hot-spots) for a considerable fraction of the time. Losses were not correlated with links with persistently high utilization, but were observed on links with low average

utilization (indicating losses due to momentarily traffic bursts). Finally, it was verified that time-of-day and day-of-week variation in link utilization exists in many Datacentres.

4.2 MapReduce

MapReduce is probably the most important Datacenter application and this is the reason we have allocated a separate section to this in this Deliverable. MapReduce clusters offer a distributed computing platform suitable for data-intensive applications (typically referred to as Big-Data). MapReduce was originally proposed by Google and it is the most widely deployed implementation. Apache Hadoop is very similar but it is open-source and is used by many companies including Yahoo!, Facebook and Twitter. MapReduce and Hadoop use a divide-and-conquer approach in which input data are divided into fixed size units processed independently and in parallel by *Map* tasks, which are executed in a distributed manner across the nodes in the cluster. After the Map phase the tasks are executed, their output is shuffled, sorted and then processed in parallel by one or more *Reduce* tasks. To support data moving in/out of the compute nodes, a distributed file system typically co-exists with the compute nodes.

A small Hadoop cluster typically comprises a single *master* and multiple *worker* nodes. It is possible to have data-only worker nodes, and compute-only worker nodes. The *Hadoop Distributed File System* (HDFS) replicates data, even on different racks. The goal is to reduce the impact of a rack power outage or switch failure, so that even if these events occur, the data may still be readable. For this purpose, the name of the rack where a worker node is, is provided by HDFS. *Job wave* is defined as the times for allocating slots for a job. In Hadoop, each Map (or Reduce) task is executed in a map (or reduce) slot. A slot is a unit of computing resources allocated for the corresponding task. A common configuration for multi-core *TaskTracker* is to set two slots for each core. If the slots required by a job are more than the available idle slots, the *JobScheduler* will first assign the available slots to allow a portion of the tasks to be loaded, forming the first "wave" in the job execution. Then, the remaining tasks are scheduled when idle slots are available, forming subsequent waves.

In [15] a two-week MapReduce workload trace was collected from Yunti, a 2000-node production Hadoop cluster. In [16] 10-months of MapReduce logs from the M45 supercomputing cluster which Yahoo! made freely available to selected universities for systems research were analyzed. This Hadoop cluster has approximately 400 nodes, 4000 processors, 3 terabytes of memory, and 1.5 petabytes of disk space. In [17] a 6-month trace from a 600-machine Facebook cluster and a 3-week trace from a 2000-machine Yahoo! cluster were collected and analyzed. In [18] 6-month storage traces from 2 Hadoop clusters at Yahoo! (with about 4100, 1900 nodes using HDFS) were analyzed.

Trace in [15] showed that, on average, each job consisted of 42 map tasks and 12 reduce tasks running on 19 nodes. The maximum number of nodes allocated to a job was observed to be 489. In [16], on average, each job consisted of 153 Map tasks and 19 Reduce tasks running on 27 nodes. The maximum number of nodes allocated to a job was 227. Map and Reduce task duration times as well as job completion times were modeled to follow the Lognormal distribution [15]-[16] (95% of the jobs completing within 20 minutes [16]). Job arrival rate were observed to follow a relatively-constant daily pattern. The number of running jobs was observed to exceed 600 at peak time [15]. Small jobs (small number of tasks per job) accounted for the majority of total number of jobs [15]-[17]. Regarding network transfer, for most of the time in-flow (data received from other nodes) and out-flow (data sent to other nodes) ranged from 10MB/s to 20MB/s [15]. As for the job waves 95% of jobs had under 5 waves of Maps and 95% of jobs had fewer than

50 waves for reduce tasks (and 80% fewer than 10 reduce waves). Traces in [16] showed that 95% of jobs had under 24 waves of Maps and 95% of jobs had fewer than 7 waves for Reduce tasks (the majority fewer than 2 reduce waves).
Regarding the distributed file system HDFS, [18] showed that workloads were dominated by the high rate of file creations/deletions. The files were very short-lived: (90% of the file deletions targeted files that were 22.27mins – 1.25 hours old). It was observed that there was a small percentage of highly popular files. Young files accounted for a high percentage of accesses, but a small percentage of bytes were stored (79% - 85% files accesses target files that were at most one day old). Finally, the observed request inter-arrivals (file openings, creations, deletions) are bursty and exhibit self-similar behavior.

5 Optically enabled data storage platforms

5.1 Storage subsystems in Datacenters - Disaggregated architectures and virtualization

Different ICT requirements must be satisfied by different configurations of the modular data storage subsystems, which form the building blocks of modern Datacenters (Figure 14a).

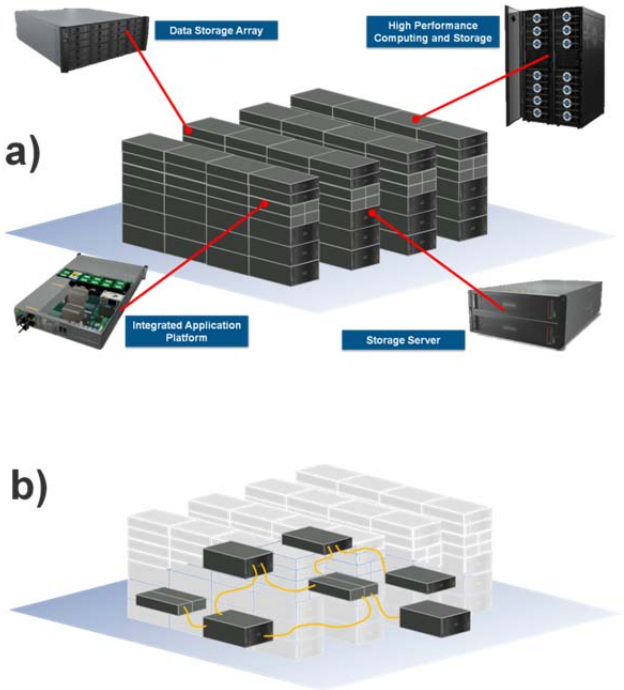


Figure 14: a) Building blocks of modern Datacenter, b) Disaggregated Datacenter architecture

These building blocks include, but are not limited to data storage arrays, integrated application platforms, storage servers, switches and high performance storage and computer subsystems.
If such building blocks could be arranged to be truly modular i.e. work in independence of their location within the Datacenter, and if interconnect length and bandwidth constraints between these subsystems could be neglected, then this would allow a disaggregated architecture as shown in Figure 14b).
In this case the combination of subsystems required to satisfy a given set of ICT requirements needn't be constrained to the same rack or cluster of racks, they could be

physically dispersed across the Datacenter. Rack scale disaggregation architectures have been proposed by Intel as part of the OpenCompute project [19].
In a similar vein, the drive to increasing virtualization of the Datacentre through Software Defined Network (SDN) architectures also promises to provide significantly greater user control, Quality of Service and flexibility while optimising resource use. Ideally the user can be provided a virtual Datacentre solution with the optimum combination and amount of compute, memory and storage, even though the actual corresponding hardware allocated could be dispersed. In order to satisfy these requirements without over-provisioning of hardware resources, one must have the capability to convey high bandwidth data over far longer distances then is typical or possible today between subsystems, and this can only be satisfied by low-cost high-bandwidth optical links. Indeed fibre-based commercial optical modules are now common in Datacenters for rack-to-rack connectivity [20]. However, while, as system bandwidths increase, the provision of ubiquitous optical links would remove the interconnect bottleneck between racks or subsystems within a rack, new bottlenecks will emerge or existing ones will become more exposed deeper in the system enclosure itself. Thus the need for commerically viable, dense interconnect solutions will continue inevitably to migrate down through the data communication tiers of the system from board-to-board, chip-to-chip and ultimately to the chip itself.

5.1.1 Migration of optical interconnect into Datacentre systems

The migration of optical connectivity within Datacentres is already underway, with hybrid electro-optical infrastructures proposed and numerous proof-of-concept technologies developed [21], [22], [23].
Notable examples include the reported deployment by IBM of optical interconnect for POWER7-IH [21] systems with 100,000s of high-performance CPU cores by leveraging dense optical transceiver and connector technologies to construct chip module optical IOs.
Fujitsu Laboratories proposed dense optical interconnect architectures for next-generation blade servers [24], with a demonstration of an electro-optical midplane with 1920 embedded optical fibres to meet the projected bandwidth requirements [25].
Xyratex, Finisar, vario-optics and Huber+Suhner demonstrated an optically enabled data storage platform, in which 12 Gb/s SAS traffic was conveyed optically between two internal controller cards along 24 PCB embedded polymer optical waveguide channels, thereby showing, for the first time, how in-system optical channels could be successfully deployed within a 12G SAS architecture [26].
HP developed an optical backplane with broadcast and MEMS based optical tapping capabilities along an embedded plastic waveguide, suitable for non point-to-point interconnect topologies [27], which was demonstrated within a proof-of-concept network switch chassis.
Commercial adoption of system embedded photonic solutions will be gated by the priority requirements relevant to the application space or market in question. Applications that prioritize performance and bandwidth density would be amongst the first adopters. For example, in the IBM Blue Waters Supercomputer, optical links are deployed at both the inter-rack and intra-rack levels [21]. In other application spaces, such as high volume ICT equipment, internal optical interconnect technologies will most likely only be adopted once commercially competitive with traditional copper interconnect solutions or once traditional interconnect can no longer meet the evolving system bandwidth requirements.

Board-mountable optical transceivers

A highly relevant development is the emergence over the past 3 years of parallel optical transceiver modules that can be mounted at any location on the PCB rather than

constrained to the card edge. This allows transceivers to be placed as close as possible to the electronic signal source (e.g. CPU, ASIC, expander) allowing electronic trace lengths and associated signal attenuation losses to be minimised and signal drive power reduced accordingly.
IBM successfully developed dense board-mounted 360 Gb/s parallel optical transceiver modules and demonstrated connectivity over embedded polymer waveguides [28], [29].
Fujitsu reported development of dense (8 x 25 Gb/s) board-mountable optical transceiver modules with microlens coupling features for electro-optical PCBs [30], [31].
Crucially, board-mounted parallel optical transceiver modules are becoming increasingly commercially available with major transceiver and connector vendors demonstrating product solutions.

Board-level optical interconnect
When considering how to embed highly dense optical channels at the board-level, there are three different key technologies at different readiness levels: 1) fibre-optic flexible laminates [32], 2) embedded planar polymer waveguides [33], [34] and 3) planar glass waveguides [35]. Each interconnect type offers different advantages making them suitable in different applications.
Laminated fibre-optic circuits, in which optical fibres are pressed and glued into place on a substrate benefit from the reliability of conventional optical fibre technology. However these circuits cannot accommodate waveguide crossings in the same layer i.e. fibres must cross over each other and cannot cross through each other. Also with each additional fibre layer, backing substrates must typically be added to hold the fibres in place, thus significantly increasing the thickness of the circuit. This would limit the long term usefulness of laminated fibre-optic circuits in PCB stack-ups. At best they can be glued or bolted onto the surface of a conventional PCB. Fibre-optic circuits were deployed in the optically enabled data storage system demonstrator developed by Xyratex as part of the PhoxTrot project [36].
Conventional step-index multimode polymer waveguides would be unsuitable for longer high speed links, in which modal dispersion would limit performance and would be equally unsuited to convey certain operational wavelengths (1310 nm or 1550 nm) over longer distances due to higher intrinsic absorbtion losses, though this can be mitigated in some polymer formulations [37]. However they would be suitable for very short reach, versatile, low cost links such as inter-chip connections on a board. They would also be suitable for applications in which certain properties of the polymer such as thermo-optic, electro-optic or strain-optic coefficients could be used to support advanced devices such as Mach-Zehnder switches or long range plasmonic interconnect. Electro-optical circuit boards with embedded polymer waveguide layers were developed by Xyratex in collaboration with IBM Research and Varioprint in 2008 [33].
Planar glass waveguide technology could combine some of the performance benefits of optical fibres, such as lower material absorbtion at longer operational wavelengths and lower modal dispersion with the ability to fabricate dense complex optical circuit layouts on single layers and integrate these into PCB stack-ups. Electro-optical circuit boards with embedded planar glass waveguides have been developed by Fraunhofer IZM and ILFA as part of the SEPIANet project [38].
Other in-system optical interconnect solutions proposed and investigated include free space optics for server backplane interconnectivity with optical WDM encoded links [39].
In order to assess the viability of embedding optical links within prevailing data storage architectures, two demonstration platforms were successfully developed and

demonstrated: 1) LightningValley was a partially optically enabled 4U data storage enclosure based on polymer optical waveguide links
2) ThunderValley2 was a fully optically enabled data storage enclosure based on embedded fibre-optic midplane and proprietary optical connectors for pluggable hard disk drives.

5.2 Optical Transmission Challenges over SAS Protocol

In order to demonstrate and validate full optical interconnectivity within a modern data storage system, the internal links to the disk drives themselves must be optically realised. This poses a greater challenge than simply linking two expander devices directly, due to the need to support the "Out-of-Band" (OOB) signaling mode in the SAS protocol. OOB is used as a means of detecting SAS / SATA links, initiating speed negotiation and polling the I/O capabilities of SAS devices on the link. Prior to the introduction of the SAS 2.1 protocol iteration, OOB involved the insertion of electrical idles of defined periods on the high speed link. Electrical idles, however cannot be reliably conveyed across a conventional optical link without additional signal processing to remove the effects of digital chatter, which would disrupt the idle periods and thus the OOB command structure. In SAS 2.1 a modified OOB scheme called "optical OOB" was introduced, in which the electrical idles were replaced by primitives, thus ensuring unbroken signal modulation at the required data rate. Although optical OOB is currently supported by most SAS devices, such as expanders and SAS controllers, hard disk drives (HDDs) and solid state drives (SSDs) do not support this signaling scheme, thus SAS links cannot be directly conveyed to a disk drive following opto-electronic conversion by a transceiver. They must pass through a buffering SAS device capable of converting from the optical OOB to OOB mode and vice versa.

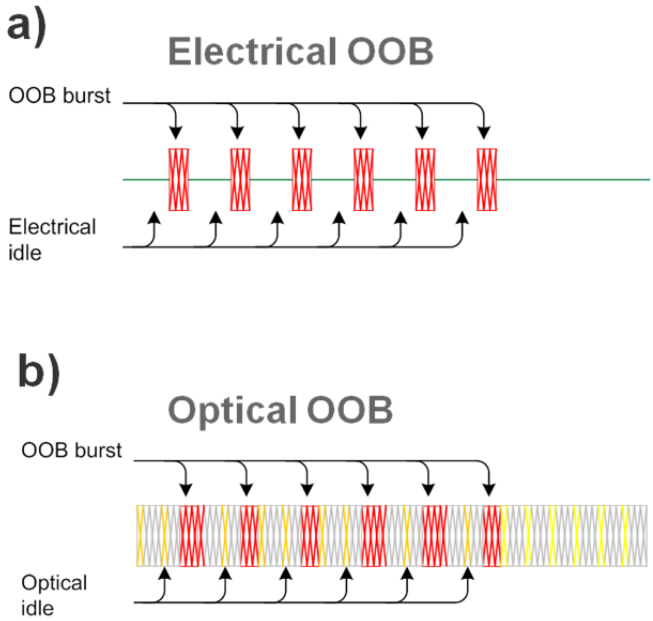


Figure 15: Out-Of-Band signalling scheme a) conventional electrical OOB, b) optical OOB scheme

5.3 LightningValley data storage platform

The "LightningValley" demonstration platform was a modified 4U OneStor™ enclosure, in which 12 Gb/s SAS traffic was conveyed optically between the SAS expanders on two pluggable internal controller cards along 24 PCB embedded polymer optical waveguide channels.

Figure 16: Partially optically enabled data storage system – LightningValley

In order to enable this, the resident SAS expander devices on each controller card had to be configured to support the Optical OOB signaling scheme and thus eliminate electrical idles on the signal path.
Each controller card housed a 48 port SAS 3 (12G) expander and 1 Finisar Board-mounted Optical Assembly (BOA) optical engine to provide the electro-optical signal conversion. The BOAs were mounted directly onto the PCB close to the expander chips to minimise the high speed electrical trace lengths and thus improve signal integrity and minimize power consumption on the electrical signal drivers. The 12 duplex channels operated at 12 Gb/s in each direction with an aggregate bandwidth of 144 Gb/s per module. As shown in Figure 17, the 12 Gb/s optical SAS data streams were thus conveyed along 12 duplex channels between the SAS expanders along 24 PCB embedded polymer optical waveguides fabricated by Vario-optics ag on both the controller cards and an electro-optical midplane to which they were connected with MT- FGAT board pluggable optical connectors.
The electro-optical midplane design was a modified version of the original 4U electrical midplane to which a 2 layer 24 waveguide flexible ribbon was attached between two Huber+Suhner MT-FGAT midplane receptacles.

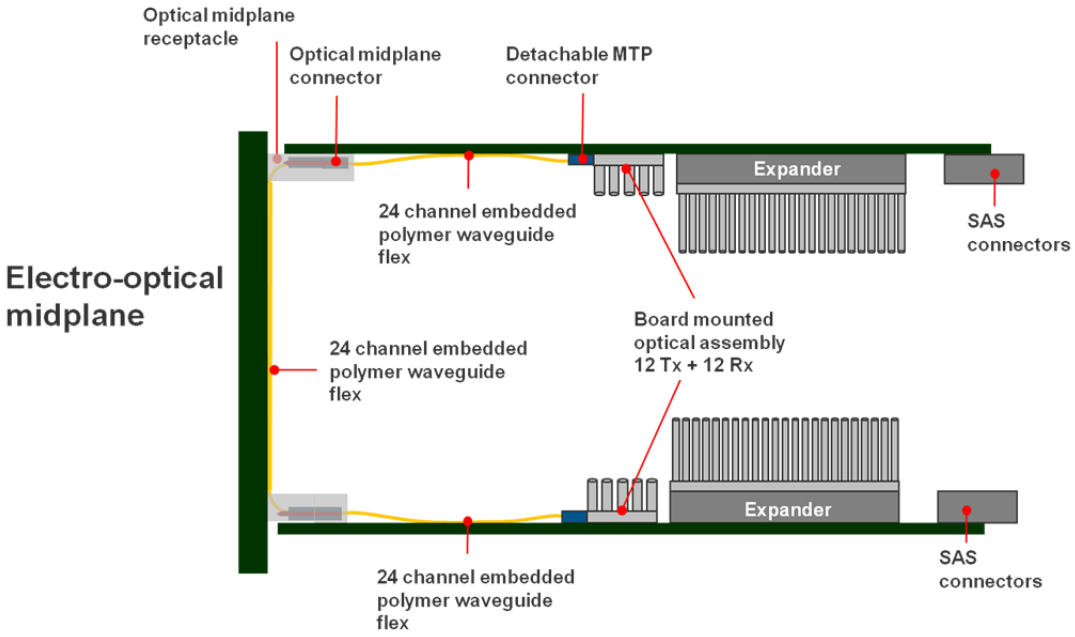


Figure 17: Schematic cross-section of SAS demonstration platform showing SAS expanders on 2 controller cards optically connected to each other via BOA optical engines, optical midplane connectors and electro-optical midplane.

The 12 bidirectional optical inter-controller card SAS links were validated with bit error rates (BER) of less than 10^{-15}, thereby showing, for the first time, how in-system optical channels could be successfully deployed within a 12G data storage system.

5.4 ThunderValley2 data storage platform

One of the early achievements during the PhoxTrot project has been the successful internal development and demonstration by Xyratex of a fully optically enabled data storage platform, in which all internal high speed links were implemented optically. This required the deployment of board-mounted optical transceivers and the development of a proprietary electro-optical midplane, pluggable optical connectors and interface cards for hard disk drives.

Figure 18: Fully optically enabled 2U24 drive data storage platform - ThunderValley2

The design of a data storage array with full internal optical connectivity was based on an existing 2U (89 mm) high, 19" wide OneStor™ 6 system enclosure Figure 18. The system includes 2 optically enabled 12G SAS controller modules, an electro-optical midplane with a full aggregate bandwidth capacity of 2.3 Tb/s and provision to optically plug 24 conventional 2.5" disk drives to the midplane.
Each controller module housed a 48 port SAS 3 (12G) expander and again 2 Finisar board-mounted optical transceiver assemblies (BOAs) used to generate the optical signals because of their compact size and high bandwidth density. They were mounted directly onto the PCB close to the expander chips to minimise the high speed electrical trace lengths and thus improve signal integrity and minimize power consumption. Each of the BOA optical transceivers consisted of 12 VCSEL based transmitters and 12 PIN based photo-detectors. The 12 duplex channels operated at 12 Gb/s in each direction with an aggregate bandwidth of 144 Gb/s per module.
The electro-optical midplane design was a modified version of the original 2U midplane in which all high speed electrical SAS signal layers had been removed and replaced by fibre-optic links in a separate thin flexible laminate, which was attached over the reduced electronic midplane PCB. Consequently the number of electronic layers in the midplane was reduced by 55% and the midboard area available for airflow increased by 20%.
The high availability interconnect topology, defined by a passive dual star configuration, requires that each disk drive supports two duplex data links on the midplane, one to each controller module. Consequently the midplane of a 24 SAS drive enclosure needs to support at least 48 duplex links (96 fibres). However in order to exploit the density advantages of optical interconnect, provision was made for 96 duplex links (192 fibres) to accommodate the possibility of quad drive interfaces in the future (four independent duplex links per drive) through the deployment of PCIe drives or enhanced SAS interfaces7. The midplane provides for each drive a small electrical connector to supply power and an optical backplane receptacle for high speed SAS signals.
Each disk drive carrier houses an interface card, which fits between the disk drive and midplane. The interface card contains a proprietary pluggable optical connector, 2 dual board-mounted optical transceivers, a small SAS expander and an electrical edge connector compliant with conventional SAS/SATA disk drive interfaces.
The proprietary parallel optical connector system comprises a plug that resides on the edge of the interface card and a receptacle that resides on the electro-optical midplane. The plug includes guiding features to allow a receiving MT ferrule from the midplane receptacle to be guided into place and connect accurately with a compliant MT ferrule in the plug section thus enabling a pluggable optical connection between interface card (and by extension the disk drive) and the electro-optical midplane. The board-mounted optical transceivers convert optical signals from the midplane to electronic signals and convey these to the small SAS expander, which, in turn, converts from the optical OOB to OOB mode supported by the disk drive.

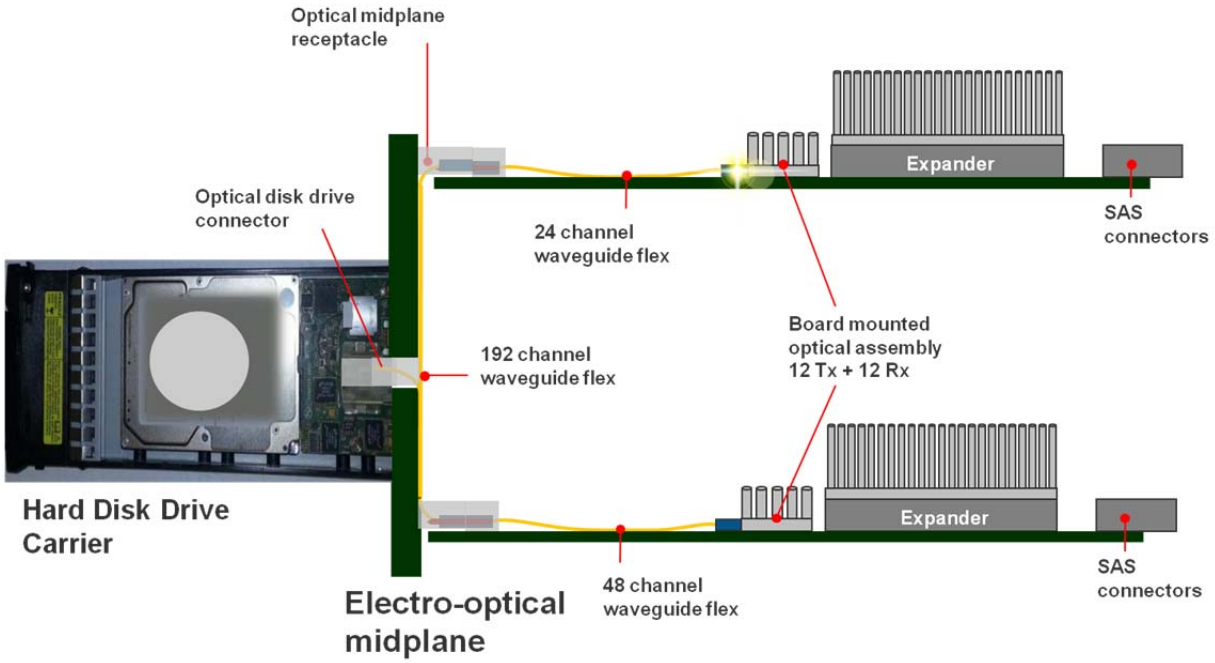


Figure 19: ThunderValley2 data flow schematic

All devices and links in the system are 12G capable with the exception of the small SAS expander mounted on the disk drive interface card, which is restricted to 6G. Therefore, although 12G optical SAS links have been successfully demonstrated between controller modules, the optical links to the disk drives could only be characterised at 6G.
The ThunderValley2 demonstration platform system has been fully validated using appropriate SAS device detection and soak-test regimes. All disk drives are detectable over the high speed optical SAS links and all SAS links have been verified with a BER of less than 10^{-15}. Comprehensive characterisation of the system is currently underway and full results will be released early next year.

6 HPC and Datacentre systems evolution

The rise and expansion of HPC and Datacenters with numerous clustered computers pose significant interconnection challenges. High- performance cluster computing configurations can take many physical configurations, which range from all equipment installed in one central area to distributed machines in multiple remote locations. Passive and in some cases active copper cables can effectively serve as short jumpers between rack levels, as well as to adjacent racks. Conventional cables in server-to-server and server-to-switch applications that are less than 10 meters in length have proven to be the most economical solution, although weight and cable density has caused maintenance problems as well as restricted cooling airflow to the equipment. Copper cables that exceed 10 meters in length and must operate at more than 10Gb/s begin to experience significant signal integrity issues. Advanced signal conditioning at the transmit and receive ends can compensate for some of this distortion, but adds cost and increases power consumption. With the development of huge Datacenters and server farms that may require the interconnection of several hundred thousand devices the issues of cable bulk, weight and performance have taken center stage. Fibre optic cables offer solutions in all three areas of concern. High-speed communication links that may stretch to a different floor or to another building on campus are natural applications for fibre optic cables. Each channel may cost more due to the electric to optic conversion process, and field termination requires skill and specialized equipment, but once installed Bit Error Rates (BER) are exceptionally low, and maintenance costs are near zero. Fibre cables are much smaller than copper equivalents thus reducing the size and weight of external cabling. Fibre cables are immune from external electro-magnetic interference and crosstalk among high-speed signals, significant advantages in dense wiring trays [40].
Figure 20 shows a comparison between copper interconnects and optical interconnects for HPC backplane.

Figure 20: Comparison Electrical I/O versus Optical I/O [41**]**

6.1 Market trends and roadmap

The need for backplanes with higher speeds, higher density (Gb/s/cm^2), less power consuming (mW/Gb/s), less cooling, low EM interference and lower cost per Gb is driven by Moore s law. Electrical solutions can still meet the law in terms of speed per differential pair, but energy consumption and cost are becoming major issues in the coming years. Performance increase
Factor 10 every 4 yrs., Exascale Systems by 2020 3 Orders increase compared to today.

Figure 21: Supercomputing performance increase [41**]**

Lately fibre optics are not only implemented for longer lengths, but are also being implemented for shorter distances. Key reasoning for this are the specific benefits of optics verses copper, power consumption, cable density, air flow (cooling), signal integrity.

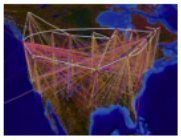
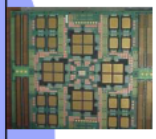
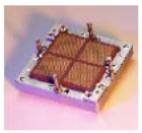
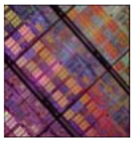
	Internet, Wide Area Network	Local Area Network	Rack-to-Rack	Card-to-Card	On-Card	On- MCM	On-Chip
							
Distance	multi-km	10 - 2000 m	30+ m	1 m	0.1 - 0.3 m	5 - 100 mm	0.1 - 10 mm
Number of lines	1	1 – 10	~100	~100-1000	~1000	~10'000	~100'000
Use of optics	Since the 80s and the early 90s	Since the late 90s	Now	2010+	2010-2015	Probably after 2015	Later, if ever

Figure 22: Fibre optic trends implemented for shorter distances [41**]**

The need for improved data signal the next hurdle has to be taken, no longer fibre optics to the front panel, but through the front panel, as near as possible to the processor. The shorter the distance the lower the signal loss and therefore higher signal integrity, which becomes more critical for higher and higher data rates. Figure 22 shows the roadmap for fibre optics integration from optical fibre at the edge only to optical interconnects integrated with the processor, last being the Phoxtrot optochip.

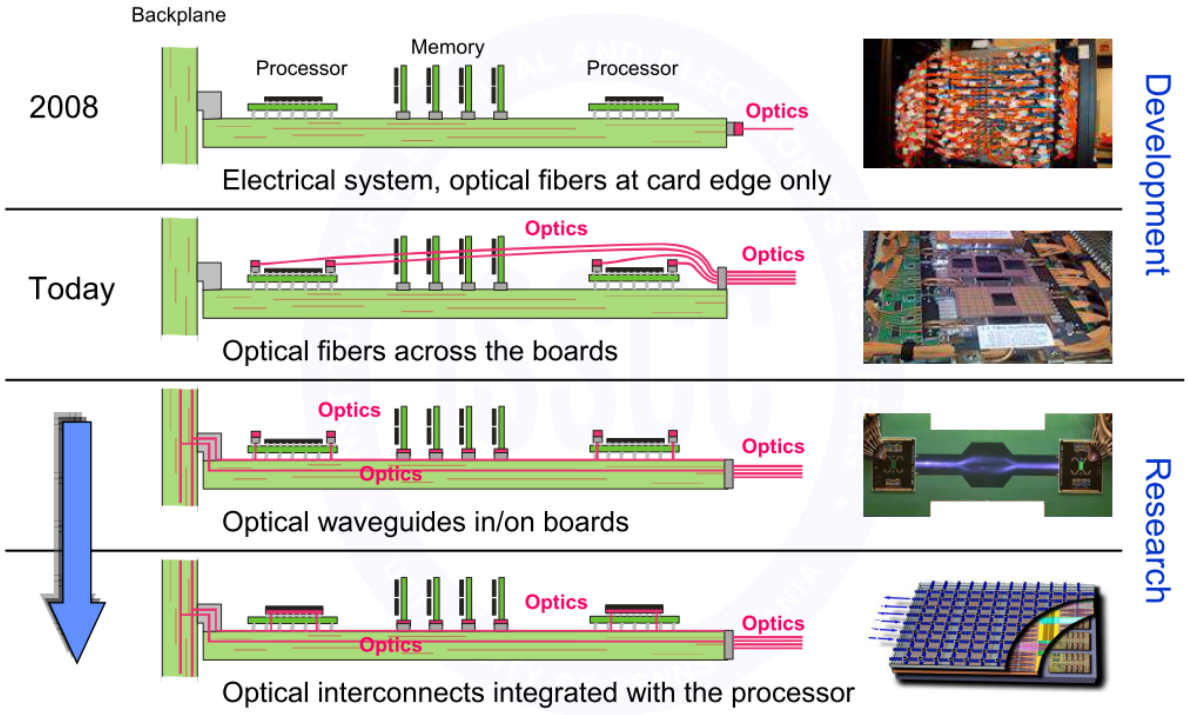


Figure 23: Fibre optics integration roadmap [41**]**

6.2 State of Art for HPC and Datacenters applications

Eflexo collaboration project between TE and IBM, general objective was to develop a functional prototype of a high-bandwidth, power efficient, small form factor, fibre- and waveguide-based transceiver platform for supercomputing/HPC, router and server applications. Main technical requirements for the demonstrator was 48 multimode channels, data rate up to 25Gb/s per channel, total link bandwidth 1.2Tb/s (0.6Tb/s up-stream and 0.6Tb/s downstream).

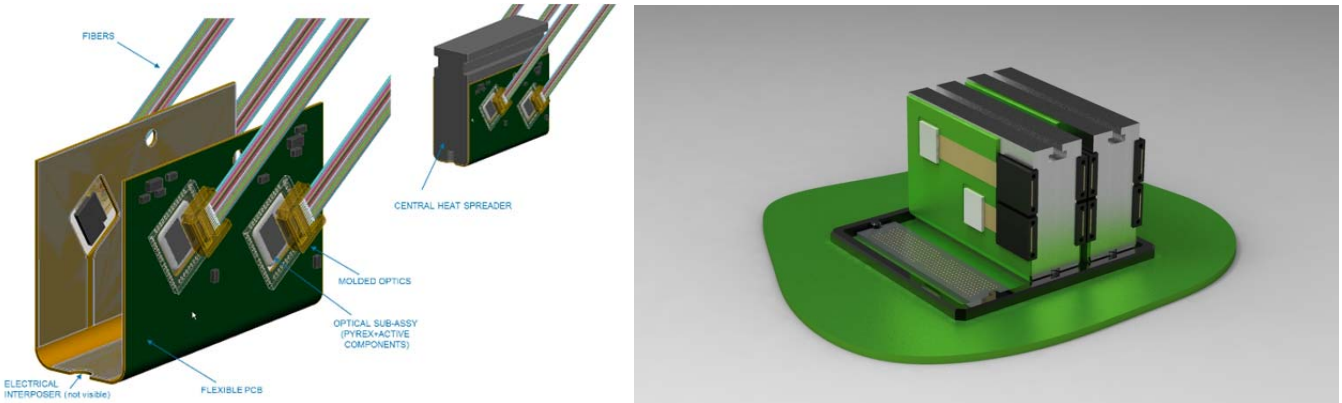


Figure 24: IBM/TE Eflexo module

7 Discussion

Generally speaking, HPC software applications exhibit well defined communication patterns, making them easier to model. An HPC application can be a fundamental "kernel" that performs a basic operation, or can be composed of a number of kernels. Ratios of MPI Communication calls, percentages of communication times (as opposed to computation times) and logical communication graphs that are available in the literature and from our experiments for HPC applications can be used to extract message sizes and communication locality models. However, the main limitation of this type of findings is that they do not convey temporal information. For example, logical communication graphs correspond to the whole execution of the application. They express volume of data (i.e. in Bytes) and not rate (bps), while transmissions are distributed over time and there might be busy and idle periods. Thus, we cannot use these models to examine network congestion but propagation latency (average low load latency for given topologies). Traces that convey temporal information are not, to the best of our knowledge, available in the literature, and related models are not readily available. We were able to execute a number of popular HPC applications and monitor the traffic that they create (traffic that crosses the switches of the system), so as to obtain traffic traces as a function of time.

Datacenters on the other hand, are multi-tenant environments where various software applications are executed, sometimes simultaneously on the same resources, and where wide variations of requirements exist. The mix of application running on the Datacenters impacts the traffic profiles. Since obtaining traces from real Datacenters is almost impossible, we performed an extensive literature review.

The ultimate goal of our work in this task is to collect information and based on that to create appropriate traffic generators for HPC and Datacenter applications that would be incorporated in the simulators that are going to be built within the framework of the Phox-Trot project. These simulators will be used to evaluate the performance of the PhoxTrot modules and overall architecture. So based on our findings in this task we can create traffic generators for some of the most popular HPC applications whose communication pattern ranges from having high locality to being very global. With respect to Datacenters, we will rely on the literature to create traffic models with respect to parameters of our interest. A traffic generator based on MapReduce/Hadoop application, which is one of the most popular Datacenter application, could also be implemented to examine the applicability of the Phox-trot solution to that.

8 References

[1] K. Asanovic, R. B. B. C. Catanzaro, J. J. Gebis, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, K. A. Yelick, M. J. Demmel, W. Plishker, S. William, and K. Yel, "The Landscape of Parallel Computing Research: A View from Berkeley," *Technical Report, UC BERKELEY*, 2006.

[2] S. Kamil, L. Oliker, A. Pinar, and J. Shalf, "Communication Requirements and Interconnect Optimization for High-End Scientific Applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 1–14, 2010.

[3] "IPM Software." .

[4] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2004.

[5] P. G. R.D. Smith, "Reference manual for the Parallel Ocean Program (POP)," *Los Alamos Unclassified Report LA-UR-02-2484*, 2002.

[6] X. S. Li and J. W. Demmel, "SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 29, no. 2, pp. 110–140, 2003.

[7] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[8] "The University of Florida Sparse Matrix Collection." .

[9] B. Alverson, E. Froese, L. Kaplan, and D. Roweth, "Cray XC Series Network."

[10] "sFlow." .

[11] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity Datacenter network architecture," *Proceedings of the ACM SIGCOMM 2008 conference on data communication - SIGCOMM '08*, vol. 38, no. 4, p. 63, Oct. 2008.

[12] T. Benson, A. Akella, and D. a. Maltz, "Network traffic characteristics of Datacenters in the wild," *Proceedings of the 10th annual conference on Internet measurement - IMC '10*, p. 267, 2010.

[13] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding Datacenter traffic characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, p. 92, Jan. 2010.

[14] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of Datacenter traffic: measurements & analysis," in *9th ACM SIGCOMM conference on Internet measurement conference (IMC)*, 2009, pp. 202–208.

[15] Z. Ren, X. Xu, J. Wan, W. Shi, and M. Zhou, "Workload characterization on a production Hadoop cluster: A case study on Taobao," in *IEEE International Symposium on Workload Characterization*, 2012.

[16] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan, "An analysis of traces from a production mapreduce cluster," in *CCGRID*, 2010, pp. 94–103.

[17] Y. Chen, A. Ganapathi, R. Griffith, and R. H. Katz, "The case for evaluating mapreduce performance using workload suites," in *MASCOTS*, 2011, pp. 390–399.

[18] C. L. Abad, N. Roberts, Y. Lu, and R. H. Campbell, "A storage-centric analysis of MapReduce workloads: File popularity, temporal locality and arrival patterns," in *IEEE International Symposium on Workload Characterization,*, 2012, pp. 100–109.

[19] Intel Corporation, "Design Guide for Photonic Architecture," 2013. [Online]. Available: http://www.opencompute.org/wp/wp-content/uploads/2013/01/Open_Compute_Project_Open_Rack_Optical_Interconnect_Design_Guide_v0.5.pdf.

[20] D. Grice, H. Brandt, C. Wright, P. McCarthy, A. Emerich, T. Schimke, C. Archer, J. Carey, P. Sanders, J. A. Fritzjunker, S. Lewis, and P. Germann, "Breaking the petaflops barrier," *IBM Journal of Research and Development*, vol. 53, no. 5, pp. 1:1– 1:16, 2009.

[21] A. Benner, D. M. Kuchta, P. K. Pepeljugoski, R. A. Budd, G. Hougham, B. V. Fasano, K. Marston, H. Bagheri, E. J. Seminaro, H. Xu, D. Meadowcroft, M. H. Fields, L. McColloch, M. Robinson, F. W. Miller, R. Kaneshiro, R. Granger, D. Childers, and E. Childers, "Optics for High-Performance Servers and Supercomputers," in *Optical Fiber Communication Conference*, 2010, p. OTuH1.

[22] K. Chen, C. Hu, K. Zheng, A. V Vasilakos, and N. Technical, "Survey on Routing in Datacenters: Insights and Future Directions," *IEEE Network*, vol. 25, no. 4, pp. 6–10, 2011.

[23] B. R. Rofoee, G. Zervas, Y. Yan, N. Amaya, Y. Qin, and D. Simeonidou, "Programmable on-chip and off-chip network architecture on demand for flexible optical intra-Datacenters.," *Optics express*, vol. 21, no. 5, pp. 5475–80, Mar. 2013.

[24] T. Miyoshi, K. Oe, J. Tanaka, T. Yamamoto, and H. Yamashima, "New System Architecture for Next-Generation Green Datacenters: Mangrove," *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL*, vol. 48, no. 2, pp. 184–191, 2012.

[25] J. Matsui, T. Yamamoto, K. Tanaka, T. Ikeuchi, S. Ide, S. Aoki, T. Aoki, and T. Ishihara, "Optical Interconnect Architecture for Servers using High Bandwidth Optical Mid-plane," pp. 3–5, 2012.

[26] Press Release, "Finisar and Xyratex Demonstrate 12G SAS Embedded Optical Interconnect Within Datacentre Subsystem at ECOC 2012." [Online]. Available: http://www.marketwatch.com/story/finisar-and-xyratex-demonstrate-12g-sas-

embedded-optical-interconnect-within-data-centre-subsystem-at-ecoc-2012-2012-09-17.

[27] AndyAtHP, "HP Networking showcases optical backplane innovation," *HP Networking*, 2011. .

[28] F. E. Doany, C. L. Schow, S. Member, B. G. Lee, R. A. Budd, C. W. Baks, C. K. Tsang, J. U. Knickerbocker, R. Dangel, B. Chan, H. Lin, C. Carver, J. Huang, J. Berry, D. Bajkowski, F. Libsch, and J. A. Kash, "Terabit / s-Class Optical PCB Links Incorporating Optical Transceivers," *Journal of Lightwave Technology*, vol. 30, no. 4, pp. 560–571, 2012.

[29] F. E. Doany, C. L. Schow, B. G. Lee, R. Budd, C. Baks, R. Dangel, R. John, F. Libsch, J. A. Kash, B. Chan, H. Lin, C. Carver, J. Huang, J. Berry, and D. Bajkowski, "Terabit / sec-Class Board-Level Optical Interconnects Through Polymer Waveguides Using 24-Channel Bidirectional Transceiver Modules," *Electronic Components and Technology Conference*, pp. 790–797, 2011.

[30] K. Tanaka, S. Ide, Y. Tsunoda, T. Shiraishi, T. Yagisawa, T. Ikeuchi, T. Yamamoto, and T. Ishihara, "High-Bandwidth Optical Interconnect Technologies for Next-Generation Server Systems," *IEEE Micro*, no. 99, pp. 1–1, 2012.

[31] T. Shiraishi, T. Yagisawa, T. Ikeuchi, S. Ide, and K. Tanaka, "Cost-effective Low-loss Flexible Optical Engine with Microlens-imprinted Film for High-speed On-board Optical Interconnection FPC Microlens imprinted film Adhesive layer Adhesive layer Polymer waveguide Mirror," pp. 1505–1510, 2012.

[32] C. Berger and M. Kossel, "High-density optical interconnects within large-scale systems," *PROCEEDINGS OF SPIE*, vol. 4942, no. 2003, pp. 222–235, 2003.

[33] R. Pitwon and K. Wang, "FirstLight: Pluggable Optical Interconnect Technologies for Polymeric Electro-Optical Printed Circuit Boards in Datacenters," *Journal of Lightwave Technology*, vol. 30, no. 21, pp. 3316–3329, 2012.

[34] L. Schares, J. A. Kash, F. E. Doany, C. L. Schow, C. Schuster, S. Member, D. M. Kuchta, P. K. Pepeljugoski, J. M. Trewhella, C. W. Baks, R. A. John, L. Shan, Y. H. Kwark, R. A. Budd, P. Chiniwalla, F. R. Libsch, J. Rosner, C. K. Tsang, C. S. Patel, J. D. Schaub, R. Dangel, F. Horst, B. J. Offrein, D. Kucharski, D. Guckenberger, S. Hegde, H. Nyikal, C. Lin, A. Tandon, G. R. Trott, M. Nystrom, D. P. Bour, M. R. T. Tan, and D. W. Dolfi, "Terabus□: Terabit / Second-Class Card-Level Optical Interconnect Technologies," *Quantum*, vol. 12, no. 5, pp. 1032–1044, 2006.

[35] H. Schröder, L. Brusberg, N. Arndt-staufenbiel, J. Hofmann, and S. Marx, "Glass Panel Processing for Electrical and Optical Packaging," *Technology*, pp. 625–633, 2011.

[36] Fraunhofer IZM, "PhoxTrot Project." [Online]. Available: http://www.phoxtrot.eu/.

[37] S. Takenobu and T. Okazoe, "Heat Resistant and Low-Loss Fluorinated Polymer Optical Waveguides at 1310/1550 nm for Optical Interconnects," *European Conference and Exposition on Optical Communications*, vol. 1, pp. 6–8, 2011.

[38] R. Pitwon and H. Schröder, "Embedded planar glass waveguide optical interconnect for Datacentre applications," *Proceedings of SPIE*, vol. 8630, no. 0, 2013.

[39] R. Rachmani and S. Arnon, "Wavelength Diversity Links," *Journal of Lightwave Technology*, vol. 30, no. 9, pp. 1359–1365, 2012

[40] Bishop Assosiates inc, "Report P-675-11 Fiber Optic Connectors in Military and Commercial Applications, Chapter 6, 2011

[41] B.J. Offrein, " Optical PCB Interconnects For Optical PCB Interconnects For Computing Applications: Computing Applications: From Niche to Mainstream From Niche to Mainstream", *ISSCC*, 2012